



**Bilkent University**  
**CS 319 Object-Oriented Software Engineering**  
**Final Report**  
**Group “Veni Vidi Code”**

**Group Members:**

- Ferhat KORKMAZ - 21901940
- Kaan TEK - 21901946
- Kerem ŞAHİN - 21901724
- Melih Fazıl KESKİN - 21901831
- Miray AYERDEM - 21901987

<b>1. Introduction</b>	<b>3</b>
<b>2. Lessons Learned</b>	<b>4</b>
<b>3. User’s Guide</b>	<b>5</b>
3.1 Admin	5
3.2 Gym Staff	5
3.3 Gym Member	6
<b>4. Build Instructions</b>	<b>7</b>
<b>5. Work Allocation</b>	<b>7</b>

## 1. Introduction

In terms of implementation, we are proud to say that we have a working application on our hands which fulfill almost all of the requirements that we stated earlier in the semester. All of the main functionalities are working without any problem:

### User Management:

- Admin Users can add two types of users: Gym Member and Gym Staff.
- All users can login and logout with their credentials.
- Different types of users see different screens and are able to perform different operations when necessary.

### Tournament:

- Gym Staff can add a Tournament by specifying necessary fields.
- Gym Staff can remove a Tournament.
- Gym Members can enroll to and disenroll from a Tournament.

### Sport Course:

- Gym Staff can add a Sport Course by specifying necessary fields.
- Gym Staff can remove a Sport Course.
- Gym Members can enroll to and disenroll from a Sport Course.

### Reservation:

- Gym Members can make a reservation by specifying necessary fields.
- Gym Staff can see all the reservations.

### Gym Program Request:

- Gym Members can request a Gym Program from a Gym Staff
- Gym Staff can view all the requests and write a Gym Program for the owner of the Gym Program Request

### Announcement

- Gym Staff can publish an Announcement.
- Both Gym Staff and Gym Member can view all the Announcements on their Home page.

We tried not to deviate too much from our initial design, and we actually did not except some negligible changes. However, there were also some minor parts that we were not able to implement in time. Notification, for example, we were planning to notify users as a result of some events. However, we preferred to focus on more important parts of the application first and in the end decided we could remove the notification functionality. We were also planning to implement a functionality that would allow some users to make payments, such as instructors or alumni. However, we again did not have enough time to successfully implement this in our code. Other than those, all other functionalities of the applications work near perfect.

## **2. Lessons Learned**

Overall we can say that we did a great job and are proud of ourselves. Everyone tried their best and fulfilled their responsibilities. However, there were of course few things that we could have done differently.

- We could have started to learn the necessary tools, programming languages, and technologies earlier. We knew this was one of the most important points and started as early as we could, however, the process of learning was somewhat slow due to other lectures' workload. Still, we could have spent our free time more efficiently and focused on learning the tools and technologies.
- The way we divided tasks between us was mainly related to front-end and back-end tasks. In other words, some of us started working on the back-end part and some of us on the front-end part, and we thought we could connect front-end and back-end in the last few days. However, making this connection was more challenging and took more time than we expected. Hence, we could have started to make this connection earlier, or one member start making this connection earlier than others while others were doing their own work.

If we are to work as a group again in the future, we are definitely going to bear in mind these few points, in order to have a more efficient and less stressful process.

### 3. User's Guide

#### 3.1 Admin

Admin is the user that is responsible for user account management. Basically, they are adding and deleting new gym member or gym staff.

- First, the Admin needs to select the “Admin” option in the log in menu from the radio buttons. After that, when the Admin enters his/her ID and password, the admin will be able to log in to the account.
- After that, the Admin will have 2 pages they can use.
- The first page the Admin can use is the Add Member section of the page. When the Admin enters the Add Member section, they will be able to see all of the users, their information.
- There will be a button that will be on the top left section, which will enable the Admin to add a user by specifying the information of the desired user.
- The second page is “Add Staff” page. This page is just the same with the Add Member page. It is dedicated for Staff and all the other functionalities are the same.

#### 3.2 Gym Staff

Gym staff is responsible for managing tournaments, reservations, courses, gym program requests, managing gym members and making announcements.

- The staff needs to select the “Gym Staff” option in the log in menu from the radio buttons. After that, when the Gym Staff enters his/her ID and password, the staff will be able to log in to the account.
- Once they are in the home page, they will be able to navigate to these pages: “Total Reservations”, “Tournaments”, “Courses”, “All Members”, “My Profile”. They can also see the announcements on this home page.
- In Total Reservations, the staff will be able to manage all the reservations that are done by the gym members. They will be able to set the status of the reservation as “Attended”. On the left part of the page, there is another button called “Set Date and Time Slots”. By using this button, the Gym Staff will be able to determine the time intervals for activities in different sports centers.
- In the Tournaments section, there is a dropdown menu which will navigate to the “Enrolled Tournaments” and “Available Tournaments”. The “Enrolled Tournaments” will show all of the enrollments to the tournaments of the users. The “Available Tournaments” section will show the Tournaments that are

currently ongoing. Also in the “Available Tournaments” part of the app, the staff will be able to add new tournaments to the system.

- In the Courses section, there is a dropdown menu “Available Courses” and “Enrolled Courses”. In “Available Courses”, the currently ongoing courses will be shown to the staff. Also, the staff will be able to add new courses with a button from the same page. In the “Enrolled Courses” section, the staff will be able to see every single user that has a course enrollment.
- In the “All Members” section of the page, the staff will be able to see every single gym members information. This page is used to restrict certain members from the sports center. By doing so, they will not be allowed to login to the system.
- In the “My Profile” section, they are able to see details about their account information they can also change their password via Change Password functionality. They can also see all the gym program requests written by the gym members and respond with a gym program by clicking one of those requests. Finally, they can publish an announcement that will appear on the home screen.

### 3.3 Gym Member

Gym member is a typical user who benefits from the sports facilities, courses and the tournaments in Bilkent University.

- The member needs to select the “Gym Member” option in the log in menu from the radio buttons. After that, when the Gym Member enters his/her ID and password, the member will be able to log in to the account.
- Once they are in the home page, they will be able to navigate to these pages: “Reservations”, “Tournaments”, “Courses”, “My Profile”, “Contact”. They can also see the announcements on this home page.
- If they decide to navigate to Reservation section, they have 2 options as Make Reservation and My Reservations. If they choose “Make Reservation”, they can make a reservation for their desired sports activity by specifying the necessary details. And in “My Reservations” section, they can see all the reservations they made previously.
- If they decide to navigate to Tournaments section, they again have 2 options as Enroll Tournament and My Tournaments. If they choose “Enroll Tournament”, they can simply enroll to a tournament among the available ones. And if they choose “My Tournaments” section, they can see the tournaments that they have enrolled.
- If they decide to navigate to Courses section, they again have 2 options as Enroll Course and My Courses. If they choose “Enroll Courses”, they can simply enroll to a sports course among the available ones. And if they choose “My Courses” section, they can see the courses that they have enrolled in.

- If they decide to navigate to My Profile section, they will be redirected to the page where they can see the details about their account and attributes. They are also able to edit their profile information upon clicking the “Edit Profile” button. They can also change their password after clicking “Change Password” button. They can also request a gym program by specifying their needs and submit the request upon clicking “Submit”. Below, they can see the latest gym program written for them by the gym staff.
- If they decide to navigate to Contact section, they can see the contact details such as name, phone number and email of all the gym staff

#### 4. Build Instructions

1. Install XAMPP. Then, start Apache and MySQL servers.
2. Clone our repository <https://github.com/MelihFazl/bilfit>.
3. Navigate to the path /bilfit/backend/bilfit from the cloned project and open IntelliJ from there. Then, please wait for the Maven indexing to finish.
4. Open “application.properties” which can be found under “src/main/resources”. Then, change the value 5551 on line “spring.datasource.url=jdbc:mysql://localhost:5551/bilfit” according to your MySQL port that can be found on XAMPP.
5. Go to the browser and type “localhost/phpmyadmin”.
6. Create a new database called “bilfit”.
7. Now, you can run “BilfitApplication.java” under “src/main/java/com.venividicode.bilfit”.
8. If everything works fine, you should be able to make a HTTP request to the server.
9. Install Postman. Then, make a GET request to the following endpoint: “<http://localhost:8080/initConstants>”. This will allow to initiate necessary entities to successfully start the application such as an admin user with id 444 and password “admin”, sport centers, and sport locations.
10. If query is successful, you should get the response as “true”.
11. On terminal, cd into the root directory of the cloned project which is simply “bilfit”. From there run the command “npm install”. Then run the command “npm start”. Note that you need to have Node.js installed on your computer in order to run “npm” commands. If you don’t have this installed, simply download it from the internet.
12. After “npm start”, you should be able to see the application running on your browser. If you cannot see it, you can visit “localhost:3000, localhost:3001, localhost:3002...”
13. If everything works fine, you should see the login page of Bilfit on your browser. Choose admin from the radio button under the input fields, and then login with the credentials given on step 9.
14. After logging in, please follow *3.1 Admin* under *3. User’s Guide* to get detailed information about adding new users.
15. If you encounter any problems, please contact us on Slack from channel *team-venividicode*

## 5. Work Allocation

- Ferhat Korkmaz

I came up with the idea to use Spring Boot as our backend service since it is compatible with Java as well as most of the OOP concepts. I worked with Kaan and Melih on the back-end side. At first, I made the user models that we exported from Visual Paradigm more compatible with Spring Boot. Then, I created repositories for necessary models that are persisted on the database, mostly all.

I created user account controllers with appropriate models such as Gym Member, Gym Staff, and Admin. During the implementation of user and admin logins, the security site was well-taught with Melih. At the end, we came up with our own tokenization idea to secure the endpoints. After that, I worked on reservation logic deeply. We had to change our design choices sometimes during the implementation of reservation. Sometimes it was caused by the technology we use, Spring Boot, sometimes, it was caused by our own fault on the reports. Also, I created two helper objects that are called PasswordHandler and IgnoredPropertyCreator. They both use Singleton design pattern since we need only one instance at a time. PasswordHandler object is used whenever a password required action takes place by using SHA256 hash algorithm, eg. create a user, change password, login. IgnoredPropertyCreator allows us to patch entities properly by ignoring new NULL values. Then, I helped the connection of front-end to back-end. I helped the following parts:

- Login
- Logout
- Local storage saves of token, id, and usertypes .
- Making Reservations.
- Listing Reservations.
- Listing Users.
- Editing Users.
- Adding Users.
- Time slot additions of a special date.

- Kaan Tek

I initially worked on the back-end part of the application with Ferhat and Melih. I mainly worked on the Announcement, Contact, GymProgram, GymProgramRequest,



Notification, and GymMemberRestriction logics. I wrote the necessary Controller, Service, and Repository classes in order to implement those logics. I also tried my best to help my friends with their implementations. I also write Javadoc comments for models, repositories, and controllers.

Once we thought we were done with the backend, we started to connect front-end and back-end. I made the connection for Announcement, Contact, Gym Program, Gym Program Request. I also worked on the My Profile page in order to allow the users to edit their profile information. I also made the connection for changing password logic.

I was also pretty active during writing the reports, like all of my friends. We divided tasks between us and I always completed my own tasks successfully and helped my friends when I could.

- Kerem Şahin
  - I worked at the front-end side of the project with Miray.
  - I have first created the navbar of the website and initiated the connections between the pages. The styling of the navbars and the separation for different users have been added.
  - I have created the dropdown menus for the navbars. DropDown Menu and hover actions have been created by me.
  - I have created styled and animated buttons that are responsive to hover.
  - I have created the Main Page of the website and I have created the Front-End part of the announcement table. In order to create the Main Page, a carousel-like component was used.
  - I have created the My Profile Page and I have done the front-end side of the edit-change password components, their dialogs. I have also created the part that enables the profile to fetch data from the current user.
  - I have created the Login Page of the website.
  - I have added Search And Pagination for certain tables that will have a lot of data in them. This will allow an easier use for the admin and gym staff to handle the big data.
  - For the connection of backend, I worked at various pages such as Course, Tournament and Reservation with my friends.
  - I have created the Gym Staff Panel so that the Gym Staff can restrict other users.
- I have created separations of pages and navbars for different users.

- Melih Fazıl Keskin

I worked on the back-end side of the project with Ferhat and Kaan. I mostly worked on the Field, Activity, SportCourse, Tournament and TournamentRegistration logics. I implemented their repositories, services and controllers. I made many changes on them until finding the best option for not suffering a problem due to nested properties. I

also changed some of the other logics in order to enhance them. For example, I updated user deletion and restriction methods to be sure about the integrity of the database. Also, I helped Ferhat about the logic of the login system.

When we started to connect our back-end with our front-end, I mostly worked on the connections of EnrollTournament, Tournaments, EnrollSportsCourses and SportCourses pages. I have added HTTP requests to the fetch methods. I made changes on both front-end and back-end sides to increase the quality and efficiency of the connections. Also I got rid of incompatibilities between the front-end and back-end on these pages. I also contributed to user restriction's connection to the database.

Other than that, I always tried to help each member in the group when they encountered a problem. I tried to find solutions with them. I always tried to actively participate in all the meetings. Also, like all of the members of my team, I have contributed to our reports by drawing UML diagrams and writing explanations.

- Miray Ayerdem
  - I worked with Kerem on frontend side.
  - MakeReservation page is created: Tables, radio buttons, and text fields are rendered according to data comes from the database and submit button saves all the inputs
  - Reservations page is created: On this page, gym members can see their own reservations information, and there are buttons for deletion. Also, staff can see all the reservations made by users, there is a button for deletion.
  - EnrollTournaments page is created: Tables that show available tournaments, and buttons that delete or edit tournament information are created for staff. Also, staff can add tournaments by writing information to text fields in the dialog. Gym members can see the available tournament and there is a button for their enrollment.
  - Tournaments page is created: On this page, users can see their enrolled tournaments as a table and there are buttons for cancelling enrollments. Staff can see all the members and their enrolled tournaments at a table. There are buttons for cancellation.
  - EnrollSportsCoursesPage is created: Tables that show available sports courses, and buttons that delete or edit sports course information are created for staff. Edit buttons open the dialog that contains text fields of information o unique sports courses. Also, staff can add sports courses by writing information to text fields in the dialog. Gym members can see the available sports courses and there is a button for their enrollment in the course.
  - Sports Courses page is created: On this page, gym members can see their enrolled sports courses as a table and there are buttons for cancelling enrollments. Staff can see all the information on courses including the participants of the courses.
  - Admin Panel page is created: On this page, the admin can see all the users and their information as a table. There are buttons for deletion of users and there are edit buttons that direct the dialog with text fields which enables change of the information of the user. Also, there is a button for adding a new user. Again, the admin should write all required information to text fields, otherwise submit button gives an alert.